# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/447,501 | 11/23/1999 | LANDY WANG | 2260 | 3903 |

7590 08/10/2005

LAW OFFICES OF ALBERT S. MICHALIK, PLLC
704 - 228TH AVENUE NE.
SUITE 193
SAMMAMISH, WA 98074

| EXAMINER | |
|---|---|
| ANYA, CHARLES E | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2194 | |

DATE MAILED: 08/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/447,501 | WANG, LANDY | |
| | Examiner | Art Unit | |
| | Charles E. Anya | 2194 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *21 January 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,4-13,15,16,27-31,35,36,38-49 and 51-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-13,15,16,27-31,35,36,38-49 and 51-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *1/21/05*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1,4-13,15,16,27-31,35,36,38-49 and 51-53 are pending in this application.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**3. Claims 1,4,5,7-9,11-13,15,27,28,31,35,36,39-46,48,49 and 51-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. 5,491,808 to Geist, Jr. in view of Developing for Windows Operating Systems (Using Driver Verifier to Expose Errors pages 1-8: hereinafter referred to as WinOS).**

4. As to claim 1, Geist teaches a method, comprising in a computer system: receiving a request from a kernel mode driver ("...allocation calls..." Col. 7 Ln. 14 - 29); determining that the kernel mode driver is to be monitored/re-vectoring the request to a kernel mode driver verifier ("... intercept..." Col. 7 Ln. 14 - 29); wherein receiving a request from a driver includes receiving a function in a kernel component of an operating system ("...actual memory management functions..." Col. 7 Ln. 40 - 46); wherein determining that the driver is to be monitored includes checking a registry setting ("...list..." Col. 9 Ln. 23 - 27); and taking action in the kernel mode driver verifier

to actively test the kernel mode driver for errors ("...value-checking..." Col. 7 Ln. 30 - 36).

5. Geist is silent with respect to the kernel mode driver verifier being capable of testing the kernel mode driver by simulating a low resource condition including failing requests for memory pool allocation.

6. WinOS teaches the kernel mode driver verifier being capable of testing the kernel mode driver by simulating a low resource condition including failing requests for memory pool allocation (page 3 lines 19 - 29).

7. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of WinOS and Geist, Jr. because the teaching of WinOS would improve the system of Geist, Jr. by not allowing a target driver to tested during memory allocation failure (WinOS page 2 paragraph 26 – 29).

8. As to claim 4, Geist, Jr. teaches the method of claim 1, wherein the request from the kernel mode driver includes a memory allocation request, and wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes allocating memory space thereto from a special pool of memory ("...ABLK list..." Col. 7 Ln. 32 - 37, Col. 8 Ln. 27 - 35, Col. 9 Ln. 64 - 67, Col. 10 Ln. 35 - 63).

9. As to claim 5, Geist, Jr. teaches the method of claim 1, wherein the request from the kernel mode driver includes a memory allocation request ("...allocation calls..." Col. 7 Ln. 18 - 21, "...function call..." Col. 8 Ln. 11 - 15).

10. Geist is silent with respect to taking action in the kernel mode driver verifier to test the driver includes marking memory bounding the memory space to detect improper access of the memory bounding the memory space.

11. WinOS teaches taking action in the kernel mode driver verifier to test the kernel mode driver includes marking memory bounding the memory space to detect improper access of the memory bounding the memory space (page 2 lines 3 – 5, page 4 lines 4 - 11).

12. As to claim 7, Geist, Jr. teaches the method of claim 1, wherein taking action in the kernel mode driver verifier to test the driver includes maintaining allocation information in at least one data structure associated with the kernel mode driver ("...ABLK list..." Col. 7 Ln. 32 - 37, Col. 8 Ln. 27 - 35, Col. 9 Ln. 64 - 67, Col. 10 Ln. 35 - 63).

13. As to claim 8, Geist, Jr. teaches the method of claim 7, wherein the request from the kernel mode driver includes a memory allocation request, and wherein maintaining allocation information includes adding data corresponding to the allocation request to the data structure ("...ABLK list..." Col. 7 Ln. 32 - 37, Col. 8 Ln. 27 - 35, Col. 9 Ln. 64 - 67, Col. 10 Ln. 35 - 63).

14. As to claim 9, Geist, Jr. teaches the method of claim 7, wherein the request from the kernel mode driver includes a memory de-allocation request ("...removing..." Col. 10

Ln. 45 - 63) and wherein maintaining allocation information includes removing data corresponding to the allocation request from the data structure (ABLK Col. 10 Ln. 45 - 63).

15. As to claim 11, Geist, Jr. teaches the method of claim 1, wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes validating call parameters ("... error checking..." Col. 10 Ln. 37 - 44).

16. As to claim 12, Geist, Jr. teaches the method of claim 1, wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes checking for resources allocated to the kernel mode driver at driver unload ("... de-allocation..." Col. 9 Ln. 64 - 67, "...removing..." Col. 10 Ln. 45 - 63).

17. As to claim 13, WinOS teaches the method of claim 1, wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes simulating a low resource condition (page 1 line 24, page 3 lines 19 – 29).

18. As to claim 15, WinOS teaches the method of claim 13, wherein simulating the low resource condition includes invalidating driver code and data (page 1 line 24, page 3 lines 19 – 29).

28. As to claim 27, Geist, Jr. teaches a computer-readable medium having computer executable instructions, comprising: receiving a request from a kernel mode driver for allocation of memory space ("...allocation calls..." Col. 7 Ln. 14 - 29, "...function call..." Col. 8 Ln. 11 - 16); determining a location of memory space to allocate ("...ABLK list..." Col. 7 Ln. 32 - 37, Col. 8 Ln. 27 - 35, Col.. 9 Ln. M - 67, Col. 10 Ln. 35 - 63); wherein determining that the kernel mode driver is to be monitored includes checking a registry setting ("...list..." Col. 9 Ln. 23 - 27) and allocating the memory space (Step 5 Col. 27 - 29, "...allocation is made..." Col. 10 Ln. 45 - 53).

19. Geist, Jr. is silent with respect to restricting access to areas bounding the location wherein any access request to at least one of the areas results in an access violation and monitoring the areas bounding the location for an access violation.

20. WinOS teaches re-vectoring the request to a kernel mode driver verifier, in the kernel mode driver verifier, testing the kernel mode driver including marking the memory bounding the memory space to detect improper access of the memory bounding the memory space/monitoring the areas bounding the location for an access violation using the kernel mode driver verifier (page 2 lines 3 – 5, page 4 lines 4 - 11) and restricting access to areas bounding the location wherein any access request to at least one of the areas results in an access violation and monitoring the areas bounding the location for an access violation (page 2 lines 3 – 5, page 4 lines 4 - 11).

21. As to claim 28, Geist, Jr. teaches the computer-readable medium of claim 27, having further compute executable instructions, comprising, instructions for detecting an access violation ("... MSG list/pool..." Col. 5 Ln. 55 - 67, Col. 7 Ln. 32 - 36, Col. 12 - 22).

22. As to claim 31, Geist, Jr. teaches a computer-readable medium having computer-executable instructions, comprising: instruction for, requesting a plurality of requests from a kernel mode driver for allocation of various distinct sets of memory ("...allocation calls..." Col. 7 Ln. 14 - 29, "...function call..." Col. 8 Ln. 11 - 16); allocating the memory (Step 5 Col. 27 - 29, "...allocation is made..." Col. 10 Ln. 45 - 53); tracking the memory allocated to the kernel mode driver on each request ("...allocation calls..." Col. 7 Ln. 14 - 29, "...function call..." Col. 8 Ln. 11 - 16);

23. WinOS teaches receiving requests for de-allocation of at least one of the sets of memory allocated to the kernel mode driver; tracking the memory de-allocated in each reallocation request; determining from the tracking whether memory remains allocated to the kernel mode driver at a time when the driver should have no memory allocated thereto; and generating an error at the if memory remains allocated (Enable memory leak detection/Enable IRP processing checking page 3).

24. As to claim 35, Geist, Jr. teaches a system for monitoring kernel mode drivers, comprising: an operating system component including an interface for receiving requests from kernel mode drivers ("...allocation calls..." Col. 7 Ln. 14 - 29, "...function

call..." Col. 8 Ln. 11 - 16); a re-vectoring component for examining the requests to determine whether requesting the kernel mode drivers are to be monitored ("...list..." Col. 9 Ln. 23 - 27).

25. Geist, Jr. is silent with reference to a driver verifier component operable connected to the re-vectoring component, the driver verifier receiving information from the re-vectoring component for a kernel mode driver that is to monitored and executing at least one test to monitor the driver, wherein in response to a request for memory from the driver, the driver verifier component allocates memory for the kernel mode driver to use from a pool of memory other than a memory pool normally allocated from when the kernel mode driver is operating unmonitored.

26. WinOS teaches a driver verifier component operably connected to the re-vectoring component, the driver verifier receiving information from the re-vectoring component for a kernel mode driver that is to monitored and executing at least one test to monitor the driver, wherein in response to a request for memory from the driver, the driver verifier component allocates memory for the kernel mode driver to use from a pool of memory other than a memory pool normally allocated from when the kernel mode driver is operating unmonitored (Driver Verifier page 1).

27. As to claim 36, WinOS teaches the system of claim 35, wherein the operating system component is a kernel component (Driver Verifier page 1).

28. As to claim 39, Geist, Jr. teaches the system of claim 35, wherein the a request from the driver includes a memory allocation request, and wherein a test by the kernel mode driver verifier includes allocating memory space thereto from a special pool of memory ("...ABLK list..." Col. 7 Ln. 32 - 37, Col. 8 Ln. 27 - 35, Col. 9 Ln. 64 - 67, Col. 10 Ln. 35 - 63).

29. Geist, Jr. is silent with respect to marking memory bounding the memory space to detect improper access of the memory bounding the memory space.

30. WinOS teach marking memory bounding the memory space to detect improper access of the memory bounding the memory space (page 2 lines 3 – 5, page 4 lines 4 - 11).

31. As to claim 40, Geist, Jr. teaches the method system of claim 35, wherein the request from the driver includes a memory de-allocation request, and wherein a test by the kernel mode driver verifier includes de-allocating the memory space and marking the de-allocated memory space to detect improper access thereof ("...de-allocation..." Col. 9 Ln. 64 - 67, "...removing..." Col. 10 Ln. 45 - 67).

32. As to claim 41, Geist, Jr. teaches the system of claim 35, wherein a test by the kernel mode driver verifier includes examining resources allocated to the driver ("... searching..." Col. 10 Ln. 45 - 63).

33. As to claim 42, Geist, Jr. teaches the system of claim 41, wherein examining resources allocated to the kernel mode driver includes tracking outstanding memory allocated to the driver (ABLK Col. 10 Ln. 45- 63).

34. As to claim 43, Geist, Jr. teaches the system of claim 41, wherein examining resources allocated to the kernel mode driver includes reviewing lists maintained by the operating system component for information therein associated with the driver ("...searching..." Col. 10 Ln. 45 - 63).

35. As to claim 44, Geist, Jr. teaches the system of claim 35, wherein a test performed by the kernel mode driver includes validating call parameters ("...error checking..." Col. 10 Ln. 37 - 44).

36. As to claim 45, Geist, Jr. teaches the method system of claim 35, wherein a test performed by the kernel mode driver includes failing requests for memory pool allocation ("...(MSG) list..." Col. 7 Ln. 34 - 36).

37. As to claim 46, Geist, jr. teaches the method system of claim 35, wherein a test performed by the kernel mode driver includes invalidating driver code and data ("...value-checking..." Col. 7 Ln. 30 - 32).

38. As to claim 48, Geist, Jr. teaches a method in a computer system for verifying system components, comprising: selecting one or more tests for verifying functionality of the system component (Col. 5 Ln. 21 - 42, "...track..." Col. 8 Ln. 11 - 16, "...monitored... " Col. 10 Ln. 1 - 7), modifying a request for system services to include execution of the selected tests/executing the modified request ("...take over..." Col. 5 Ln. 43 – 54, "...intercept..."Col. 7 Ln. 14 - 29, "...thunk..." Col. 7 Ln. 42 - 67).

39. Geist is silent with respect to one of the tests includes restricting access to a resource such that an attempted access to the resource causes an access violation; and generating errors for any test failures.

40. WinOS teaches one of the tests includes restricting access to a resource such that an attempted access to the resource causes an access violation; and generating errors for any test failures (page 2 lines 3 – 13, page 4 lines 4 - 11).

41. As to claim 49, Geist, Jr. teaches the method of claim 48, wherein the system component comprises a device driver (NLM Col. 9 Ln. 9 -13).

42. As to claim 51, Geist, Jr. teaches the method of claim 48, further comprising applying a test condition designed to detect a specific error ("...errors..." Col. 5 Ln. 32 - 42).

43. As to claim 52, WinOS teaches the method of claim 51, wherein applying the test condition includes restricting available system resources (page 4 lines 4 - 11).

44. As to claim 53, see the rejection of claim 48 above.

**45. Claims 6,29 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,491,808 to Geist, Jr. in view of Developing for Windows Operating Systems (Using Driver Verifier to Expose Errors pages 1-8: hereinafter referred to as WinOS) as applied to claim 7 above, and further in view of U.S. Pat. No. 5,689,707 to Donnelly.**

46. As to claim 6, Geist, Jr. teaches the method of claim 1, wherein the request from the driver includes a memory de-allocation request (" ...removing..." Col. 10 Ln. 45 - 67).

47. Geist, Jr. is silent with respect to taking action in the kernel mode driver verifier to test the kernel mode driver includes marking de-allocated memory space to detect improper access of the de-allocated memory space.

48. Donnelly teaches taking action in the kernel mode driver verifier to test the driver includes marking de-allocated memory space to detect improper access of the de-allocated memory space ("...free() function..." Col. 8 Ln. 10 - 26).

49. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Donnelly, WinOS and Geist, Jr. because the teaching of Donnelly would improve the system of WinOS and Geist, Jr. by providing a means for efficiently detecting and managing memory allocation/de-allocation (Donnelly Col. 2 Ln. 42 - 67).

50. As to claim 29, Geist, Jr. teaches the computer-readable medium of claim 27, having further computer-executable instructions, comprising, instructions for receiving a request from the kernel mode driver for de-allocation of the memory space ("...de-allocation..." Col. 9 Ln. 64 - 67, "...removing..." Col. 10 Ln. 45 - 63).

51. Geist, Jr. is silent with respect to restricting access to de-allocated memory space, wherein any access request to the de-allocated memory space results in an access violation and monitoring the de-allocated memory space for an access violation.

**52.** Donnelly teaches restricting access to de-allocated memory space, wherein any access request to the de-allocated memory space results in an access violation and monitoring the de-allocated memory space for an access violation ("...track..." Col. 4 Ln. 34 - 67).

53. As to claim 30, see the rejection of claim 29 above.

**54. Claims 10 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,491,808 to Geist, Jr. in view of Developing for Windows Operating Systems (Using Driver Verifier to Expose Errors pages 1-8: hereinafter referred to as WinOS) as applied to claim 7 above, and further in view of PCT WO 95/22104 to Parker et al.**

55. As to claim 10, Geist, Jr. is silent with respect to the method of claim 7, further comprising: providing the allocation information to a user interface.

56. Parker teaches the method of claim 7 further comprising providing the allocation information to a user interface (Step 190 page 20 lines 3 - 34).

57. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Parker, WinOS and Geist, Jr. because the teaching of Parker would improve the system of WinOS and Geist, Jr. by providing a means for alerting a user of memory allocation problem (Parker page 20 lines 10 - 16).

58. As to claim 38, Geist, Jr. is silent with respect to the system of claim 35, further comprising: a user interface for writing driver information to the registry.

59. Parker teaches the system of claim 37, further comprising: a user interface for writing driver information to the registry ("... Save prompt..." page 21 lines 6 - 23).

**60. Claim 16 and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. 5,491,808 to Geist, Jr. in view of Developing for Windows Operating Systems (Using Driver Verifier to Expose Errors pages 1-8: hereinafter referred to as WinOS) as applied to claim 1 above, and further in view of U.S. Pat. No. 6,430,665 B1 to Allison et al.**

61. As to claim 16, Geist, Jr. is silent with respect to the method of claim 1, wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes checking for timers, in de-allocated pooled memory.

62. Allison teaches the method of claim 1, wherein taking action in the kernel mode driver verifier to test the kernel mode driver includes checking for timers, in de-allocated pooled memory (figure 4 Col. 4 Ln. 18 - 67).

63. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Allison, WinOS and Geist, Jr. because the teaching of Allison would improve the system of WinOS and Geist, Jr. by preventing changes to the configuration of memory while memory allocator is performing the sorting function (Allison Col. 4 Ln. 60 – 64).

64. As to claim 47, see the rejection of claim 16 above.

***Conclusion***

65. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat. No. 5,355,469 to Sparks et al.: Invention directed to automatic detection of errors in computer program caused by erroneous memory allocations and de-allocations.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E. Anya whose telephone number is (571) 272-3757. The examiner can normally be reached on M-F (8:30-6:00) First Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, An Meng-Ai can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles E Anya
Examiner
Art Unit 2194

cea.

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER